

Universiti Teknologi MARA

**Image Based
Static Hand Gesture Recognition
As
User Input
Using
Self Organizing Maps**

Fakhzan Firdaus Abdullah

**Thesis submitted in fulfillment of the requirements for
Bachelor of Science (Hons) Intelligent Systems
Faculty of Information Technology And
Quantitative Sciences**

November 2006

APPROVAL
IMAGE BASED
STATIC HAND GESTURE RECOGNITION
AS
USER INPUT
USING
SELF ORGANIZING MAPS
BY
FAKHZAN FIRDAUS ABDULLAH

This thesis was prepared under the and approval of thesis advisor and supervisor, Pn. Marina Binti Yusoff, Department of Intelligent Systems. It was submitted to the School of Information Technology and Quantitative Sciences and Quantitative Science and was accepted in partial fulfillment of the requirements for the degree of Bachelor of Science Honors Intelligent System.

Approved by:

.....
Pn Marina Binti Yusoff
Thesis Supervisor, Advisor

Date: 2 November, 2006

Abstract

Gesture interaction is one of the aspects of HCI that has gathered a lot of attention in the recent years. Apart from the mouse, no significant HCI technologies have garnered attention in the consumer market. Gesture in humans for example are movements or symbols that are made by using any part of the human body that acts as way of communication. This project is focusing on the development of a symbolic hand gesture recognition prototype based the shape of the hand using Neural Network by means of the Kohonen's Self Organizing Map (SOM) to be used as computer input. For this project, several sets of hand gestures are collected from different people. The images are preprocessed by scaling, cleaning and converting into a format that is easy to be input into the neural network, namely binary. In the neural network design, each of the bits that represents the image will be input into the SOM network. The SOM method of identification is by reducing the dimensions of inputs and grouping similar patterns together. Therefore, by adjusting the weights of each neuron path, the images used to train the network will group into similar pattern groups. Testing is then done by inputting a different set of images from the trained images and identify whether the correct gesture is identified. After the identification, commands to the Windows console are executed according to the recognized image that is associated with the gesture. In the preprocessing side, the system has proven to normalize images quickly and accurately. However, in the SOM Network, with the lack of trained images, the accuracy rate of the system is at most 25%, with variable results on each training. Nonetheless, since the system is using a viable and tested Kohonen engine, the system can be improved with a more extensive collection of images for training. Moreover, albeit the lack of accuracy, each of the programs associated with the gestures are executed flawlessly.

Table of Contents	PAGE
DECLARATION	ii
ACKNOWLEDGEMENT	iii
ABSTRACT	iv
CONTENTS	v - vii
LIST OF TABLES	viii
LIST OF FIGURES	ix
LIST OF KEYWORDS	x

CHAPTER 1: INTRODUCTION

1.0. Introduction	1
1.1. Problem Statement	2
1.2. Objectives	2
1.3. Scopes	2
1.4. Significance of the Project	3

CHAPTER 2: LITERATURE REVIEW

2.0. Introduction	4
2.1. Human-Computer Interaction	4
2.1.1. Principals of HCI	5
2.1.2. Basic Interaction and Application of HCI	6
2.2. Gesture	7
2.2.1. Gesture Recognition as Computer Access	9
2.2.2. American Sign Language	10
2.2.3. Image Recognition using Neural Networks	11
2.3. Self Organizing Maps	12
2.4. Summary	14

CHAPTER 3: RESEARCH APPROACH AND METHODOLOGY

3.0. Introduction	15
3.1. Knowledge Acquisition	17
3.1.1. Data Collection and Image Acquisition	18
3.2. Image Preprocessing	19
3.3.1 Cleaning	19
3.3.2 Normalization	20
3.3. Design and Implementation of the Kohonen Self Organizing Map	22
3.3.1. How a Kohonen Network Recognizes	23
3.3.1.1. The Structure of the Kononen Neural Network	23
3.3.1.2. Input Normalization	25
3.3.1.3. Calculating Each Neuron's Output	26
3.3.1.4. Mapping to Bipolar	26
3.3.1.5. Choosing the Winner	27
3.3.1.6. How a Kohonen Network Learns	28
3.3.1.7. Learning Rate	29
3.3.1.8. Adjusting Weights	30
3.4.2. Prototype	31
3.4.3. Network Topology and Parameters	31
3.4.4. Training and Testing	32
3.4.5. Automatic Program Launching	32
3.4. Documentation	33
3.5. Software and Hardware Requirement	34
3.6. Summary	34

CHAPTER 4: CONSTRUCTION

4.0. Introduction	35
4.1. Prototyping Phase	35
4.1.1. Technology	35
4.1.2. Implementation	36
4.1.2.1. Prototype	37